Annals of Operations Research

A HEURISTIC ALGORITHM BASED ON MULTI-ASSIGNMENT PROCEDURES FOR NURSE SCHEDULING

--Manuscript Draft--

Manuscript Number:	ANOR-1756R3
Full Title:	A HEURISTIC ALGORITHM BASED ON MULTI-ASSIGNMENT PROCEDURES FOR NURSE SCHEDULING
Article Type:	SI: PATAT 2010
Keywords:	Nurse Scheduling Problem, Assignment Problem, Heuristic Algorithms, Combinatorial Optimization.
Corresponding Author:	Ademir Aparecido Constantino, Ph. D. State Universty of Maringá Maringá, Paraná BRAZIL
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	State Universty of Maringá
Corresponding Author's Secondary Institution:	
First Author:	Ademir Aparecido Constantino, Ph. D.
First Author Secondary Information:	
Order of Authors:	Ademir Aparecido Constantino, Ph. D.
	Dario Landa-Silva, Ph. D.
	Everton Luiz de Melo, Mc.
	Candido Ferreira Xavier de Mendonça, Ph. D.
	Douglas Rizzato
	Wesley Romão, Ph. D.
Order of Authors Secondary Information:	
Abstract:	This paper tackles a Nurse Scheduling Problem which consists of generating work schedules for a set of nurses while considering their shift preferences and other requirements. The objective is to maximize the satisfaction of nurses' preferences and minimize the violation of soft constraints. This paper presents a new deterministic heuristic algorithm, called MAPA (multi-assignment problem-based algorithm), which is based on successive resolutions of the assignment problem. The algorithm has two phases: a constructive phase and an improvement phase. The constructive phase builds a full schedule by solving successive assignment problems, one for each day in the planning period. The improvement phase uses a couple of procedures that re-solve assignment problems to produce a better schedule. Given the deterministic nature of this algorithm, the same schedule is obtained each time that the algorithm is applied to the same problem instance. The performance of MAPA is benchmarked against published results for almost 250,000 instances from the NSPLib dataset. In most cases, particularly on large instances of the problem, the results produced by MAPA are better when compared to best-known solutions from the literature. The experiments reported here also show that the MAPA algorithm finds more feasible solutions compared with other algorithms in the literature, which suggest that this proposed approach is effective and robust.

1 Introduction

In this paper, we tackle a *Nurse Scheduling Problem* (NSP) which consists of assigning work shift patterns to a team of nurses over a pre-defined scheduling period in such a way that nurses' preferences (soft constraints) for what type of shift to work in each day are best satisfied while additional requirements (hard constraints) are met. A penalty cost is associated to the non-satisfaction of nurses' preferences and also to the non-satisfaction of the additional requirements. Thus, the objective is to generate feasible nurse schedules with a minimum total penalty cost. The general nurse scheduling problem was classified by Osogami and Imai (2000) as NP-hard. In the literature, we find many different descriptions and models for nurse scheduling due to the different characteristics and policies that arise in each hospital. Similarly, we can find a wide variety of solution procedures to tackle the nurse scheduling problems and a fair comparison between the many proposed algorithms seems to be impractical as discussed by Maenhout and Vanhoucke (2007).

Cheang et al. (2003) and Burke et al. (2004) provide surveys of nurse scheduling problems and solution approaches. These surveys reveal that most of the heuristic algorithms for nurse scheduling algorithms in the literature are based on local search procedures. Even recent works tackling nurse scheduling in a multi-objective fashion (e.g. Burke et al. 2012) are still largely based on local search. The distinctive feature of the heuristic algorithm proposed here is that it is based on exact resolution of successive assignment problems instead of local search. The surveys by Cheang et al. (2003) and Burke et al. (2004) also identify the need for a set of benchmark problem instances to facilitate the comparison of the many proposed algorithms for the problem. Towards this, Vanhoucke and Maenhout (2005) proposed a large dataset called NSPLib, which also includes a problem instance generator. NSPLib has 248,640 nurse scheduling problem instances randomly generated and they are classified according to their size and complexity. A subset of these instances is called the 'realistic' set which includes instances with a scheduling period of 28 days. The other set is called the 'diverse' set which includes instances with a scheduling period of 7 days. Instances of both types are used in the experiments of this paper. As mentioned above, NSPLib includes a program for generating different tests instances by changing the type of

contract (full-time or part-time), skill sets, etc. For a detailed description, see Vanhoucke and Maenhout (2005). The NSPLib problem instances are available at: http://www.projectmanagement.ugent.be/nsp.php. In their work on nurse scheduling using the NSPLib dataset, Maenhout and Vanhoucke (2006; 2007; 2008) have proposed several algorithms and reported a range of results.

Other benchmark datasets for nurse scheduling problems have been made available more recently. For example, the First International Nurse Rostering Competition 2010 (see Haspeslagh et al. 2012 and http://www.kuleuven-kulak.be/nrpcompetition for details) includes 60 problem instances classified in three groups according to the expected computational difficulty. Also, Tim Curtois at the University of Nottingham maintains a large collection of employee scheduling benchmark problem instances including nurse scheduling (see http://www.cs.nott.ac.uk/~tec/NRP/ for details). In addition De Causmaecker and Vanden Berghe (2011) proposed a classification system for nurse rostering problems, comparing three datasets: http://www.cs.nott.ac.uk/~tec/NRP/ (Burke et al. 2008), http://allserv.kahosl.be/~burak/project.html (Bilgin et al. 2008) and NSPLib at http://www.projectmanagement.ugent.be/nsp.php (Vanhoucke and Maenhout 2005). In their attempt to classify and compare the problem instances in these tree datasets, they proposed and discussed several notations and categories. According to the authors, the advantage of NSPLib is its large size, which facilitates statistical analysis of different solution approaches.

Developing formal models for the many specific objectives and constraints in nurse scheduling problems and applying optimization methods to solve them are very difficult tasks. Then, developing heuristic algorithms to tackle these problems is a common and effective approach. In fact, Vanhoucke and Maenhout (2005) suggest that the purpose of NSPLib is to be a benchmark dataset for evaluating heuristic approaches to solve nurse scheduling problems. The best results for the NSPLib instances have been obtained with different meta-heuristic approaches including the Electromagnetic method by Maenhout and Vanhoucke (2007), Scatter Search by (Maenhout and Vanhoucke, 2006) and Genetic Algorithms by Maenhout and Vanhoucke (2008). The present paper presents a new deterministic heuristic algorithm called MAPA (multi-assignment problem-based algorithm), which produces new best solutions for some instances in NSPLib.

According to Cordeau et al. (2002) a good heuristic must satisfy some criteria such as *simplicity*, *flexibility*, *accuracy* and *speed*. They also state that "*algorithms that contain too many parameters are difficult to understand and unlikely to be used*". The MAPA algorithm proposed here is simple because it does not require parameter tuning and it uses the well-known linear assignment problem that is solvable in polynomial time. It is flexible because it is well suited to tackle different constraints (hard and soft) by only adapting the procedure to calculate the matrix of costs (see Section 4.1). It also has reasonable accuracy and speed which is illustrated by the experiments described in the next sections.

The remainder of this paper is as follows. The problem description is given in Section 2. A high-level description of the proposed MAPA algorithm is given in Section 3 and then a detailed description is provided in Section 4. Experimental results are presented and discussed in Section 5. The final Section 6 draws overall conclusions and suggestions for future research.

2 Description of the Nurse Scheduling Problem

The nurse scheduling problem addressed in this paper is the same as stated by Maenhout and Vanhoucke (2007) with test instances from the NSPLib. The problem involves requirements that must be met (hard constraints) and requirements that are desirable to meet (soft constraints) when constructing the schedule. Hard constraints in this problem are the prohibition of certain successive shift assignments to nurses (for example a night shift followed by an early or a day shift), maximum number of consecutive assignments of the same type (i.e. identical shift assignments), minimum and maximum number of overall working assignments for a nurse and minimum number of consecutive assignments of the same type (i.e. identical shift assignments). Soft constraints in this problem are the minimum coverage requirement (to satisfy the workload demand of each day) and the nurses' preferences.

Nurses express their preferences for the shifts that they want to work in each day. A cost is associated to every shift and this cost is inversely proportional to the expressed preference, i.e. less preferred shifts carry a higher cost. The cost of violating hard constraints is added to the cost of violating soft constraints to obtain the total solution cost which should be minimized. Full details of the costs calculation are given in Section 4 when the MAPA algorithm is described.

More formally, the Nurse Scheduling Problem tackled here can be stated as follows. A set of nurses N needs to be scheduled within a scheduling period of d_{max} days (d=1, ..., d_{max}). Each nurse needs to be assigned to a set of shifts in the scheduling period while minimizing the cost of violating hard and soft constraints. Thus, we have:

N: set of nurses, index n (n=1, ...,n_{max}), n_{max}=|N|;

D: set of days within the scheduling period, thus d_{max}= |D|;

S_d: set of required shifts for day d, index s (s=1, ...,s_d), s_d=|S_d|.

The term *shift* refers to a given working period (early, day or night shift) or a rest period (free shift), although the starting/ending times of each shift are not defined in the NSPLib instances. Note that S_d represents the minimum coverage requirement, i.e. |S_d| is the minimum number of nurses required on day d, then |S_d|≤|N|. A *duty roster*, or *roster*, is a sequence of shifts assigned to one nurse during the scheduling period of d_{max} days. A solution or nurse schedule is a collection of n_{max} duty rosters.

3 A Multipartite Model for Nurse Scheduling

In this paper we represent the above nurse scheduling problem as an acyclic multipartite graph with d_{max}+1 partitions, where the first partition of vertices corresponds to the set of nurses and the remaining partitions correspond to the sets of shifts (i.e. one partition per day in the scheduling period). Figure 2 shows a sample of this representation in the case where n_{max}=4 nurses. An edge represents a possible assignment of a shift to a nurse in a particular day (according to the partition number). There are no edges connecting vertices in the same partition. Instead, a sequence of edges connecting vertex n from the first partition (corresponding to nurse n) to a vertex in the last partition indicates the sequence of shifts that are assigned to nurse n. The weight associated to an edge is the cost of assigning a particular shift to nurse n according to the nurse's preferences.

More formally, let's have a graph G=(T, A), where T is the set of vertices and A is the set of edges as described above. The set T is composed by the partitions T_o, T_1, T_2,..., T_{dmax}, where T_o is the set of vertices representing the nurses and T_d (d from 1 to d_{max}) is the set of vertices representing the shifts on day d. Thus, we have a multipartite graph representation. The objective is to find n_{max} paths from the first to the last partition while minimizing the total cost. Each path

represents a duty roster for one nurse, i.e. the sequence of shifts assigned to a nurse for each day of the scheduling period. In order to find these paths we propose a heuristic algorithm that solves successive assignment problems, each one corresponding to a matching problem between two consecutive partitions (bipartite graph). This assignment problem is formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n_{\max}} \sum_{j=1}^{n_{\max}} c_{ij}^{d} \cdot x_{ij}^{d} \tag{5}$$

$$\text{Subject to:} \quad \sum_{i=1}^{n_{\max}} x_{ij}^{d} = 1, \quad j = 1,\ldots,n_{\max} \tag{6}$$

$$\sum_{j=1}^{n_{\max}} x_{ij}^{d} = 1, \quad i = 1,\ldots,n_{\max} \tag{7}$$

$$x_{ij}^{d} \in \{0,1\}, \quad i = 1,\ldots,n_{\max};\ j = 1,\ldots,n_{\max} \tag{8}$$

The cost matrix c_{ij}^{d} is always a square matrix of size n_{max}^{2} and has different interpretation and structure depending on the algorithm phase, as explained in the next section. In some cases, the cost c_{ij}^{d} in (5) is the cost of edge (*i*, *j*) connecting partitions T_{d-1} and T_d, where index *i* corresponds to a nurse or roster, while the index *j* can be a shift or a roster. In other cases, the cost c_{ij}^{d} is the cost of replacing a shift *j* in the duty roster of nurse *i*. Note that $c_{ij}^{d} = \infty$ if there is no edge (*i*, *j*). The binary decision variable x_{ij}^{d} indicates an assignment or not of vertex *i* to vertex (nurse) *j*. Constraints (6) and (7) indicate a one-to-one assignment between two partitions. This means that each nurse (partition T_0) will be assigned exactly one (working or free) shift for each partition (day). The main idea is to find the minimal cost matching for each bipartite graph so that we find the n_{max} paths (each path corresponds to an individual nurse roster). The main advantage of tackling the nurse scheduling problem in this way is that the assignment problem can then be solved in polynomial time using the algorithm proposed by Carpaneto and Toth (1987) which has a polynomial running time complexity of $O(n_{max}^{3})$. Also, the heuristic procedure is deterministic producing the same solution every time is applied to the same problem instance. However, note that in our approach we need to solve the assignment problem many times in order to obtain a full nurse schedule.

4 The Proposed Heuristic Algorithm

We propose a multi-assignment problem-based algorithm (MAPA) which consists of two phases, both based on successive resolutions of an assignment problem between two consecutive partitions in the multipartite graph described above. In the first phase, an initial solution (set of duty rosters) is built. In the second phase, two procedures are employed to improve the initial solution by modifying the previous assignments between the partitions.

4.1 Construction Phase

The construction phase starts by generating the multipartite graph as defined in Section 3. An initial solution is obtained by solving d_{max} successive assignment problems from the first to the last day of the scheduling period.

As stated above, the square matrix of costs c_{ij}^{d} has different interpretations in each phase of the algorithm. In this first phase c_{ij}^{d} is the cost of assigning shift j to nurse i on day d. We note that in the nurse scheduling problem instances tackled here, the number of nurses available to work on a day is usually greater than or equal to the number of required working shifts on that day (covering requirement), i.e. $|S_d| \leq |N|$ as stated in Section 2. Then, we complete the cost matrix with *spare shifts* in order to get a square matrix c_{ij}^{d} where a *spare shift* is a type of shift considered in the problem (early, late, night or free shift). This means that the algorithm can assign more working shifts than needed in day d (further discussion below on how we deal with this). In this first phase, the matrix c_{ij}^{d} is divided into two blocks as shown in Figure 1.

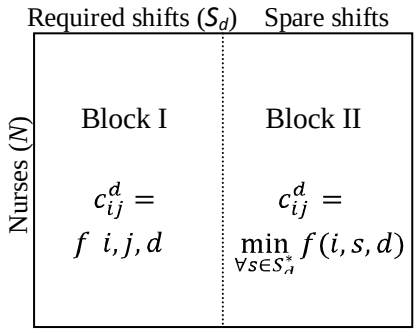

Figure 1: The cost matrix structure for the assignment of shifts to nurses, Block I ensures the cover requirement and Block II contains the *spare shifts* needed to form a square cost matrix.

Block I contains the shifts that satisfy the required coverage on day *d* and Block II contains the *spare shifts* added to form a square matrix c_{ij}^{d} where the number of available nurses on day *d* is greater or equal than the number of nurses required in the coverage. Since the minimum coverage requirement is guaranteed by the shifts in Block I, any assignment of *spare shifts* in Block II to nurses is permitted, including the assignment of free shifts. The function for calculating the costs in Block I is defined as follows:

$$f\ i,j,d\ = pc\ i,j,d\ + P_h.nHCV + P_s.nSCV \tag{9}$$

where $pc\ i,j,d$ is the penalty cost (related to the nurse's preferences) for assigning shift *j* to nurse *i* on day *d*; $nHCV$ is the number of hard constraint violations due to this assignment; P_h is the penalty for the violation of a hard constraint; $nSCV$ is the number of soft constraint violations due to this assignment and P_s is the penalty for the violation of a soft constraint. This cost function is as proposed by Maenhout and Vanhoucke (2007).

Let S_d^* be all the required shifts in S_d including free shifts, then $\mathrm{S_d^*} = \mathrm{S_d} \cup$ free shifts . Therefore, the equation $c_{ij}^{d} = \min_{\forall s \in S_d^*} f(i,s,d)$ in Block II gives the following information: the penalty cost of assigning spare shift *j* to nurse *i* and the shift type in S_d^* that will be assigned to this nurse *i* as a spare shift. Note that the value of c_{ij}^{d} in Block II is the same for nurse *i*. Each cost c_{ij}^{d} in Block II is taken as the minimum cost among the c_{ij}^{d} costs in Block I for the corresponding nurse, also considering the assignments of free shifts to that nurse. This means that for nurse *i*, each of the costs in Block I corresponds to an assignment (early, day, night or free shift) towards a covering of the required shifts in the workload while the corresponding costs in Block II are equal to the minimum of the costs in Block I for that same nurse. Since the assignments in Block II correspond to spare (not required shifts) our approach produces schedules that definitely meet the minimum coverage requirements and possibly exceed that requirement for some days in the scheduling period. Hence, the associated constraint violation costs are set accordingly to complete the overall multi-assignment problem.

An assignment problem is constructed and solved for each day of the scheduling period. Note (see Figure 2) that in the first assignment of shifts (day 1) from partition 1 to partition 2 there is no previous assigned shift. However, from

the second assignment (day 2) onwards, the previous assignments must be considered when calculating the cost matrix. That is, when calculating the c_{ij}^{d} cost for nurse *i* on day *d*, the shifts assigned to that nurse in previous days are taken into account. In order to calculate c_{ij}^{d} a simple procedure (called *constraints update*) checks the sequence of shifts assigned to nurse *i* in the previous days to day *d*. The procedure checks the constraints, e.g. if the minimum/maximum number of working assignments is satisfied or not. The time spent in calculating c_{ij}^{d} depends on the length of that sequence of shifts which is known to be not greater than d_{max}. This process is repeated for each day in the scheduling period. Then, at the end of this multi-assignment process, we have constructed an initial solution, i.e. a duty roster for each nurse. The construction phase just explained is expressed in the following pseudo-code (AP stands for assignment problem).

```
Procedure Construction
Begin
  For d=1 to d_max do:
    Construct the cost matrix c_ij^d for day d;
    Solve the AP corresponding to the cost matrix c_ij^d ;
    Assign the shifts to the nurse according to the AP solution;
End.
```

4.2 Improvement Phase

The improvement phase is composed of two procedures that aim to improve the initial solution obtained in the construction phase. The first procedure, called *Cutting and Recombination Procedure* (CRP), performs successive '*cuts*' in the multipartite graph before each day *d*. This means dividing the duty roster in two parts (left- and right-hand sides) and then constructing another assignment in the *cut* made, as it is shown in Figure 3. Therefore, a new assignment problem is formulated with new square costs matrix c_{ij}^{d} and then solved after each *cut*. An important difference when solving this new assignment problem is that c_{ij}^{d} represents the cost of assigning to nurse *i* on day *d*, the left-hand side of schedule *j* to the right-hand side of the same schedule which takes into account the shifts already assigned before and after the *cut*. In order to calculate this cost, the algorithm explores which spare shifts (those with the minimum cost) can be *updated* (reassigned) for the nurse in such a way that the new reassignment has a reduced cost. Such *updates* in the assignment of spare shifts after the cut are

possible due to the degree of flexibility in the nurse' preferences. The satisfaction of such preferences takes into account the left and right-hand sides of the cut schedule, which might be different from the construction of the initial schedule when there is no assignment to the right of the given partition.

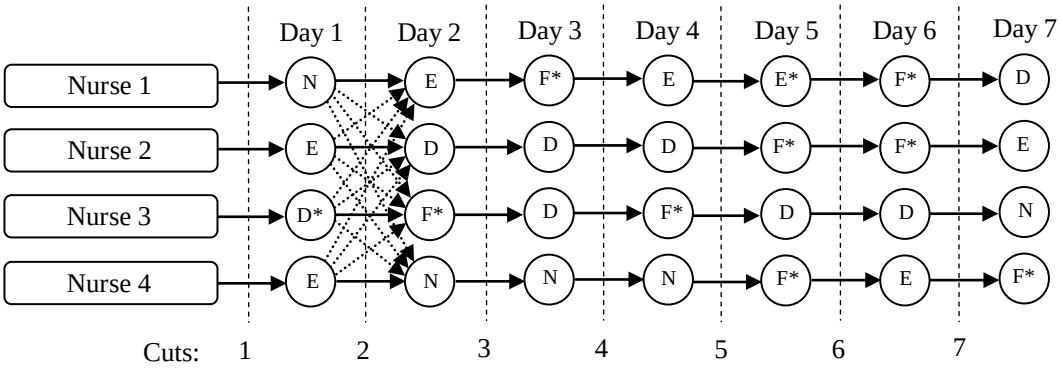

Figure 2: Example of a multipartite graph for 4 nurses and 7-days scheduling period, showing a duty roster with a cut before day 2 and possible recombinations (dashed lines) of partial rosters. Letters E, D, N, F mean an Early, Day, Night and Free shift, respectively; * means a spare shift.

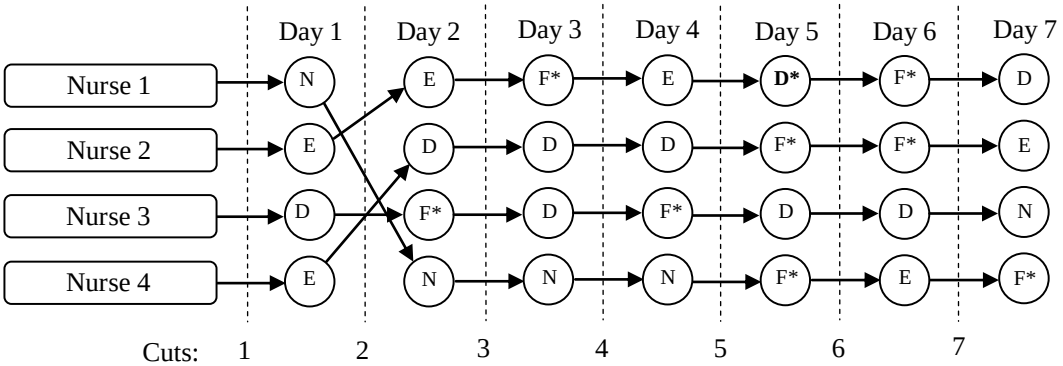

Figure 3: Example of reassignment after the cut (Figure 2 and then solving the new assignment problem) with the cutting and recombination procedure (CRP), resulting in a reassignment of working and spare shifts. Note that on day 5 a spare shift was changed (updated) for reducing the cost corresponding to nurse 2 individual roster (assuming that nurse 2 prefers shift D to shift E).

The pseudo code of the CRP improving procedure is given below (AP stands for assignment problem).

```
Algorithm CRP
Begin
   For d=1 to d_max do:
     Construct the matrix c_ij^d after performing a cut before day d;
     Solve the AP corresponding to the cost matrix c_ij^d ;
     Reassign left- and right-hand sides of the schedule according
     to the AP solution;
     Update the spare shifts for each nurse roster to reduce the
     overall solution cost;
End.
```

The second improvement procedure, called *Shift Redistribution Procedure* (SRP), aims to decrease the total cost of the solution by redistributing shifts among nurses in each day as shown in Figure 4. Since the solution cost is associated to the nurses' preferences, the same shift assigned to different nurses may contribute with different costs to the overall schedule cost. Then, this SRP improving procedure consists of selecting a day (partition) in the schedule and then reassigning the n_{max} shifts on this day to the n_{max} rosters. The cost of each association is an element of the matrix c_{ij}^{d} , where c_{ij}^{d} is the cost of replacing shift *j* in day *d* of the schedule for nurse *i*. This calculation of the costs is analogous to the one performed in the CRP procedure and involves the minimum cost of the spare shifts as well as the *constraints update* procedure described in Section 4.1.

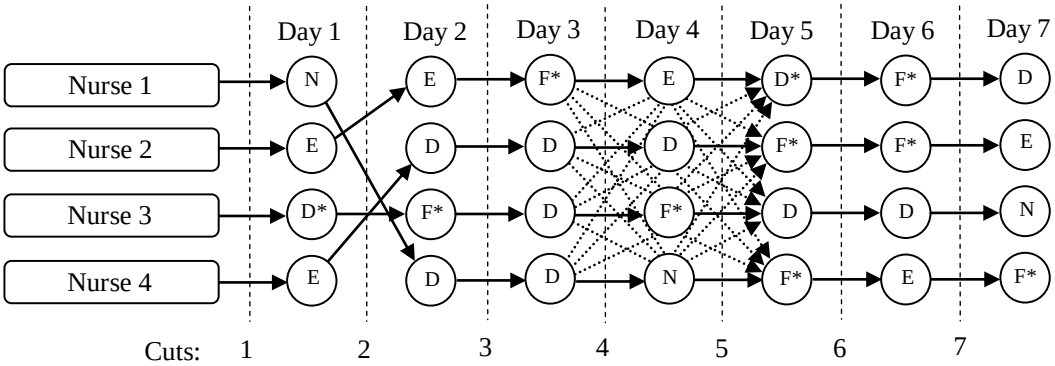

Figure 4. Example of reassignment in the shift redistribution procedure (SRP). New possibilities of shift association on day 4 are represented by dashed arrows.

Once the cost matrix is generated and the related assignment problem is solved, the current solution is altered through shift exchanges and some spare shifts may be replaced. Figure 5 shows an example of such alteration.

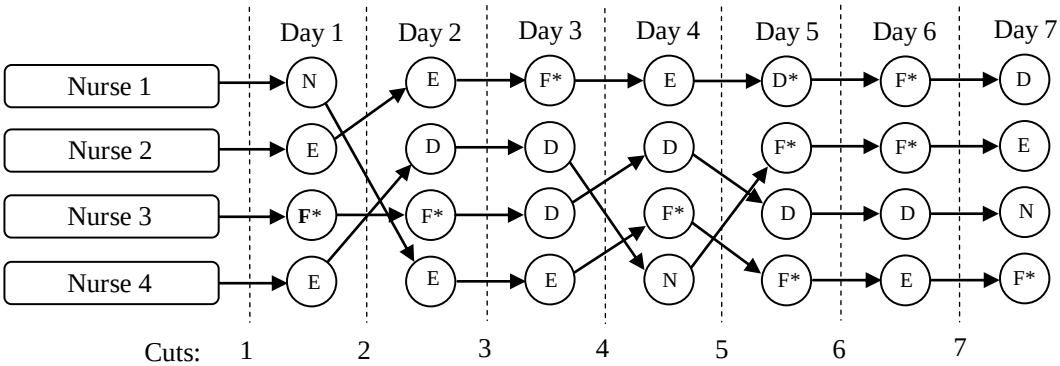

Figure 5: Example of shift exchange. Individual rosters after shift redistribution on day 4 including the change (update) of a spare shift on day 1 for nurse 3 (from shift D to shift F).

This SRP improving procedure is repeated for all partitions (all days) according to the pseudo-code shown below (AP stands for assignment problem).

```
Algorithm SRP
Begin
   For d=1 to d_max do:
     Construct the matrix c_ij^d to replace the shifts on day d;
     Solve AP corresponding to cost matrix c_ij^d ;
     Replace the shifts in the rosters according to AP solution;
     Update the spare shifts for each nurse roster to reduce the
     overall solution cost;
End.
```

The two improving procedures CRP and SRP described above are performed in a sequential fashion in both directions covering the d_{max} partitions: *forward* (d=1 to d_{max}) and *backward* (d=d_{max} downto 1). The procedures are performed until there is no improvement for a certain number of iterations (*NumIt*). Therefore, we defined four variants: RCP_Forward(*s*), SRP_Forward(*s*), RCP_Backward(*s*) and SRP_Backward(*s*), where *s* represents a solution (full schedule). Let *Val*(s) be the cost of solution *s*, which is equal to the objective function value of the last assignment problem solved, then the overall proposed improvement phase in our algorithm works as shown in the pseudo-code below (the fixed execution order of the improvement procedure variants was decided by preliminary experimentation). The parameter *NumIt* is the predefined number of times that the whole improvement procedure is attempted without an improvement in the current solution.

```
Procedure Improvement(s)
Begin
   count:=0;
   Repeat
      s':=s;
      s':=RCP_Forward(s');
      s':=SRP_Forward(s');
      s':=RCP_Backward(s');
      s':=SRP_Backward(s');
     if Val(s')=Val(s) then
        count:=count + 1
      else
        count:=0;
   until count=NumIt;
end.
```

5 Performance Analysis of MAPA

5.1 Experimental Setting

The proposed MAPA algorithm was implemented in Pascal programming language and the tests were performed on a PC with two 3.2 GHz quad-core Xeon processors and 16GB of RAM running Windows XP. The problem instances were obtained from the NSPLib library (Maenhout and Vanhoucke, 2005).

We tested MAPA on 248,640 problem instances split in two groups: Group 1 with 233,280 instances involving 1-week schedules and Group 2 with 15,360 problem instances involving 4-week schedules. In the Group 1 we find 29,160 requirement-costs problem instances involving a scheduling period of 7 days (1-week schedule). These instances are divided by problem size: 25, 50, 75, and 100 nurses, each subset containing 7,290 instances. Each problem instance has a different set of requirements per day and different preference costs. Furthermore, there are also 8 cases with different preferences and coverage constraints. Then, each of these 8 preferences-coverage cases may be combined with each of the 29,160 requirements-costs problem instances, forming a total of 233,280 1-week schedule problem instances. In the Group 2 we find 1,920 requirement-costs problem instances involving a scheduling period of 28 days (4-week schedule). These instances are divided by problem size: 30 and 60 nurses, each subset containing 960 instances. Again, we combine the 8 preference-coverage cases with each of the 1,920 requirement-costs problem instances forming a total of 15,360 4-week schedule instances. There are two important issues we must discuss about the use of NSPLib. The first issue is that the results we obtained for 38 of these instances (33 instances with 30 nurses and 5 instances with 60 nurses) could not be compared to existing results. We believe that the solution costs are misreported in the NSPLib because in some cases the reported cost is less than zero, which is not possible considering the given definition of penalty costs. The penalty values for soft constraint violations used here are the same as the ones used by Maenhout and Vanhoucke (2006) and Maenhout and Vanhoucke (2007), i.e. $P_h = P_s = 100$. Also, we fixed $NumIt=3$ in the improvement phase.

The second issue in using NSPLib is that the minimum coverage constraint (working shifts required in each day) is always satisfied by our algorithm (as explained in Section 4.1), but this is not the case in some of the (infeasible)

solutions reported in the NSPLib. In other words, our MAPA procedure satisfies the minimum coverage hard constraint, while some of the solutions reported in NSPLib satisfy some of the constraints but not necessarily the coverage constraint. We followed exactly the same definition stated by Maenhout and Vanhoucke (2007), i.e. "*a nurse schedule is said to be feasible if the coverage constraints and all other case-specific constraints are satisfied".* Then, given this issue with feasibility in some solutions reported in NSPLib, in order to compare our results to those NSPLib infeasible solutions, we made the following adjustments. At the end of the improvement phase, if a solution is infeasible we apply a procedure that changes shifts to attempt satisfying all hard constraints except the coverage constraints. Then, if a required working shift is not assigned, a penalty is added to the solution cost. However, if another hard constraint is satisfied, then a penalty is deducted from the solution cost. For example, if a nurse works more than the maximum allowed number of working days, this constraint violation can be satisfied by replacing a working shift with a free shift (in case of a spare shift). Anyway, the solution stays infeasible, but is more comparable to the solutions reported in NSPLib.

5.2 Results and Discussion

We compare the results obtained by MAPA to the results reported in NSPLib. These results are split in two groups, one for the 1-week instances and the other for the 4-week instances. Each group is then split according to the problem size, i.e. the number of nurses.

The top four sections of Table 1 show the results reported by NSPLib and the results obtained by MAPA for the 233,280 1-week instances involving 25, 50, 75 and 100 nurses. The two sections of Table 1 below the double lines report results for the 15,322 (not 15,360)[1] 4-week instances involving 30 and 60 nurses. The best results are highlighted in bold and the data given in each column is as follows. Column one gives the number of nurses $|N|$. Column two gives a label for each case (instances of the same type). Column three gives the total number of solved instances (#Inst) for each case. Columns four and six give the average solution cost (AvgCost) reported in NSPLib and obtained by MAPA respectively.

[1] We excluded 33 cases involving 30 nurses and 5 cases involving 60 nurses for which NSPLib reports infeasible solutions.

Columns five and seven give the average number of constraints violations (AvgVl) reported in NSPLib and obtained by MAPA respectively. Columns eight and nine give the relative difference between the NSPLib results and MAPA results with respect to the average solution cost (GpCost) and the number of constraint violations (GpVI) respectively. The last three columns give the percentage of times in which the best solution cost is reported in NSPLib (column ten), obtained by MAPA (column twelve) or there is a tie (column %both).

The %GAP value is calculated as follows:

$$\%\text{GAP} = (Val(\text{MAPA}) - Val(\text{NSPLib})/Val(\text{NSPLib}))\times 100 \quad (10)$$

where *Val*() is the solution cost value obtained by the given method.

The results shown in Table 1 indicate that MAPA performed poorly on the 1-week small instances (with 25 nurses), performed better on the 1-week larger instances (with 50, 75 and 100 nurses), but performed very well on the 4-week instances (with 30 and 60 nurses). In the 4-week instances MAPA always reached better results than those reported in NSPLib. Looking at the overall performance of MAPA compared to the solution costs reported in NSPLib across all 1-week schedules, we can report that MAPA obtained solutions with better average cost on 7.26% of the instances. However, when considering all 4-week schedules, MAPA obtained solutions with better average cost on 99.48% of the instances. We highlight case 15 with 60 nurses where MAPA showed its best performance, that is, a 12.21% lower average cost solutions with 32.40% fewer constraint violations. Case 15 for 30 nurses is also a case where MAPA performed very well.

Table 2 shows the average solution cost for those instances in which NSPLib reports feasible solutions (recall that NSPLib reports infeasible solutions for some instances). This table shows the number of instances for which a feasible solution is reported both by MAPA and NSPLib (#BothFeas), the number of instances for which a feasible solution is reported by MAPA or by NSPLib (#Feas). Note that MAPA and NSPLib do not always report feasible solutions for the same number of problem instances.

The results shown in Table 2 indicate that MAPA reached better solutions and also more feasible solutions on larger instances, mainly 1-week schedules with 100 nurses and 4-week schedules with 30 and 60 nurses. We highlight that on the 4-week schedules MAPA obtained more feasible solutions in all cases.

Table 1: Comparing the results (solution cost and number of soft constraint violations) obtained by MAPA to the results reported in NSPLib.

$\|N\|$	Case	#Inst	NSPLib		MAPA		%GAP		%BestSol		
			AvgCost	AvgVl	AvgCost	AvgVl	GpCost	GpVl	NSPLib	%both	MAPA
25	1	7,290	305.11	0.530	306.25	0.530	0.37	0.00	**46.28**	47.94	5.78
	2	7,290	293.82	0.530	294.34	0.530	0.18	0.00	**25.93**	69.66	4.42
	3	7,290	321.99	0.538	323.48	0.538	0.46	0.03	**58.26**	32.28	9.47
	4	7,290	303.26	0.530	303.97	0.530	0.24	0.00	**33.51**	59.66	6.83
	5	7,290	336.89	0.711	339.37	0.715	0.74	0.52	**65.79**	29.70	4.51
	6	7,290	294.81	0.530	295.32	0.530	0.17	0.00	**25.45**	69.77	4.79
	7	7,290	408.74	1.250	441.59	1.548	8.04	23.84	**83.40**	13.59	3.00
	8	7,290	330.90	0.719	335.69	0.753	1.45	4.77	**52.47**	39.03	8.50
50	1	7,290	587.07	0.848	587.44	0.848	0.06	0.00	**27.52**	51.22	21.26
	2	7,290	565.07	0.848	565.24	0.848	0.03	0.00	13.66	68.57	**17.76**
	3	7,290	615.58	0.868	615.53	0.869	-0.01	0.03	27.72	36.32	**35.95**
	4	7,290	583.68	0.848	583.84	0.848	0.03	0.00	18.74	58.93	**22.33**
	5	7,290	670.28	1.429	672.91	1.443	0.39	1.04	**42.15**	36.90	20.95
	6	7,290	567.41	0.848	567.43	0.848	0.00	0.00	12.15	65.17	**22.67**
	7	7,290	829.02	2.730	870.87	3.125	5.05	14.49	**64.72**	20.34	14.94
	8	7,290	652.73	1.400	660.34	1.473	1.16	5.19	26.80	39.45	**33.74**
75	1	7,290	912.86	1.503	912.15	1.503	-0.08	-0.01	16.45	40.69	**42.87**
	2	7,290	888.31	1.503	888.07	1.503	-0.03	-0.01	9.47	58.33	**32.21**
	3	7,290	954.41	1.524	952.80	1.521	-0.17	-0.18	17.34	32.41	**50.25**
	4	7,290	902.16	1.503	901.68	1.503	-0.05	0.00	11.33	50.27	**38.40**
	5	7,290	1,004.27	2.029	1,005.13	2.037	0.09	0.39	28.38	33.06	**38.56**
	6	7,290	889.69	1.503	889.44	1.503	-0.03	0.00	9.67	58.05	**32.28**
	7	7,290	1,214.34	3.671	1,284.07	4.362	5.74	18.82	**55.24**	21.59	23.17
	8	7,290	993.65	2.067	997.98	2.119	0.44	2.52	16.05	31.43	**52.52**
100	1	7,290	1,389.23	1.665	1,387.28	1.663	-0.14	-0.11	10.08	32.55	**57.37**
	2	7,290	1,346.80	1.663	1,346.01	1.663	-0.06	-0.02	6.79	43.48	**49.73**
	3	7,290	1,468.56	1.704	1,464.12	1.691	-0.30	-0.75	9.90	23.50	**66.60**
	4	7,290	1,375.60	1.664	1,373.98	1.663	-0.12	-0.09	7.04	34.50	**58.46**
	5	7,290	1,540.01	2.602	1,541.29	2.618	0.08	0.61	21.80	25.24	**52.96**
	6	7,290	1,349.82	1.663	1,348.84	1.663	-0.07	-0.03	6.61	41.54	**51.85**
	7	7,290	1,870.16	5.172	1,938.01	5.825	3.63	12.63	**50.07**	17.53	32.40
	8	7,290	1,513.95	2.569	1,520.31	2.646	0.42	3.00	13.83	21.69	**64.49**
30	9	959	1,911.806	4.024	1,861.785	3.923	-2.62	-2.51	1.04	0.31	**98.64**
	10	960	1,821.199	3.924	1,806.778	3.919	-0.79	-0.13	4.79	1.15	**94.06**
	11	957	2,016.964	4.134	1,938.501	3.931	-3.89	-4.90	0.52	0.10	**99.37**
	12	960	1,857.499	3.924	1,837.518	3.919	-1.08	-0.13	1.67	0.73	**97.60**
	13	959	2,030.919	4.668	1,930.881	4.217	-4.93	-9.67	1.98	0.00	**98.02**
	14	960	1,837.875	3.942	1,822.353	3.931	-0.84	-0.26	4.27	0.94	**94.79**
	15	951	2,473.512	8.231	2,208.909	5.839	-10.70	-29.06	7.68	0.00	**92.32**
	16	941	2,022.393	5.149	2,010.258	4.964	-0.60	-3.59	14.03	0.53	**85.44**
60	9	960	3,786.042	7.020	3,675.269	6.741	-2.93	-3.98	1.15	0.00	**98.85**
	10	960	3,610.247	6.769	3,567.293	6.741	-1.19	-0.42	1.15	0.10	**98.75**
	11	960	3,984.298	7.217	3,819.042	6.741	-4.15	-6.60	1.04	0.00	**98.96**
	12	960	3,681.692	6.765	3,627.718	6.741	-1.47	-0.35	0.31	0.21	**99.48**
	13	960	4,015.435	8.190	3,799.254	7.243	-5.38	-11.56	0.83	0.00	**99.17**
	14	960	3,644.343	6.814	3,596.639	6.758	-1.31	-0.81	0.94	0.10	**98.96**
	15	960	4,875.376	14.758	4,280.155	9.976	-12.21	-32.40	3.85	0.00	**96.15**
	16	955	4,003.423	8.825	3,917.626	8.422	-1.99	-4.57	10.83	0.00	**89.17**

Table 2: Comparing the results (solution cost and number of feasible solutions) obtained by MAPA to the results reported in NSPLib.

$\|N\|$	Case	#Inst	# BothFeas	NSPLib		MAPA	
				AvgCost	# Feas	AvgCost	# Feas
25	1	7,290	6,435	**250.553**	6,435	251.394	6,435
	2	7,290	6,435	**239.395**	6,435	239.689	6,435
	3	7,290	6,421	**266.482**	6,421	267.677	**6,422**
	4	7,290	6,435	**248.629**	6,435	249.094	6,435
	5	7,290	6,261	**263.472**	6,261	265.228	6,261
	6	7,290	6,435	**240.368**	6,435	240.637	6,435
	7	7,290	5,642	**279.050**	**5,839**	282.044	5,642
	8	7,290	6,228	**256.453**	**6,241**	257.495	6,232
50	1	7,290	6,563	**499.941**	6,563	500.020	6,563
	2	7,290	6,563	478.054	6,563	**477.905**	6,563
	3	7,290	6,534	526.071	6,537	**525.694**	**6,544**
	4	7,290	6,563	496.466	6,563	**496.306**	6,563
	5	7,290	6,215	**523.088**	6,215	523.641	**6,221**
	6	7,290	6,563	480.358	6,563	**480.069**	6,563
	7	7,290	5,570	**547.861**	**5,707**	549.278	5,574
	8	7,290	6,217	508.347	**6,233**	**508.060**	6,225
75	1	7,290	6,466	757.929	6,466	**756.830**	6,466
	2	7,290	6,466	733.380	6,466	**732.795**	6,466
	3	7,290	6,442	797.099	6,442	**795.464**	**6,454**
	4	7,290	6,466	746.826	6,466	**746.000**	6,466
	5	7,290	6,274	795.008	6,274	**794.510**	**6,276**
	6	7,290	6,466	734.754	6,466	**734.133**	6,466
	7	7,290	5,648	**834.904**	**5,795**	835.540	5,654
	8	7,290	6,244	779.549	**6,253**	**778.138**	6,252
100	1	7,290	6,597	1,217.768	6,597	**1,215.337**	**6,600**
	2	7,290	6,599	1,175.595	6,599	**1,174.085**	**6,600**
	3	7,290	6,563	1,292.882	6,563	**1,289.106**	**6,588**
	4	7,290	6,597	1,203.919	6,597	**1,201.737**	**6,600**
	5	7,290	6,290	1,269.268	6,290	**1,267.558**	**6,309**
	6	7,290	6,598	1,178.512	6,598	**1,176.831**	**6,600**
	7	7,290	5,706	**1,334.991**	**5,797**	1,335.186	5,729
	8	7,290	6,299	1,246.684	6,309	**1,243.779**	**6,323**
30	9	959	659	1,476.656	659	**1,443.880**	**668**
	10	960	669	1,404.157	669	**1,390.821**	669
	11	957	653	1,576.562	653	**1,524.534**	**666**
	12	960	667	1,439.109	667	**1,420.769**	**669**
	13	959	638	1,524.378	638	**1,477.876**	**657**
	14	960	668	1,418.090	668	**1,403.674**	**669**
	15	951	590	1,613.158	592	**1,579.397**	**631**
	16	941	621	1,488.361	621	**1,477.105**	**628**
60	9	960	664	3,015.901	664	**2,948.224**	**675**
	10	960	673	2,875.618	673	**2,838.235**	**675**
	11	960	658	3,199.853	658	**3,098.603**	**675**
	12	960	673	2,945.602	673	**2,897.960**	**675**
	13	960	653	3,117.025	653	**3,011.757**	**670**
	14	960	670	2,902.136	670	**2,862.655**	**674**
	15	960	634	3,321.891	634	**3,197.087**	**657**
	16	955	646	3,048.249	646	**2,999.334**	**656**

The results shown in Tables 1 and 2 give us some evidence that the multiple resolutions of the assignment problems in each step of the improvement procedures constitute an effective approach to build larger schedules. Also, these results indicate that the improvement phase is particularly useful when making reassignments of shifts for nurses by targeting existing costly assignments.

5.3 Computational Time

Table 3 shows the average computational time taken by MAPA and the corresponding computational time reported in NSPLib. Without taking into account that the machines used were different, the last column in the table gives an indication of the difference in computation time between MAPA and NSPLib.

Table 3: Computation time consumed by MAPA and computational time reported in NSPLib.

|*N*|	|*D*|	Case	#Inst	Average time (seconds)		%GAP of time
				NSPLib	MAPA	
25	7	1 a 8	58,320	2.162	0.718	-66.780
50	7	1 a 8	58,320	5.212	2.825	-45.809
75	7	1 a 8	58,320	11.641	6.834	-41.291
10	7	1 a 8	58,320	21.623	13.629	-36.970
30	28	9 a 16	7,647	22.102	92.246	317.368
60	28	9 a 16	7,675	61.906	447.035	622.119

Note that for smaller instances the average execution time of MAPA is shorter than the time reported in NSPLib. As the size of the instance grows, the running time of the proposed MAPA method becomes larger compared to the time reported in NSPLib. This also indicates that although the proposed multi-assignment approach is very effective in finding low-cost feasible solutions for large instances, the computational efficiency of MAPA is an aspect that could be improved. The resolution of each assignment problem is done in polynomial time, but the number of assignment problems solved together with the improvement phase, slow down the method on larger instances.

5.4 Performance of the Improvement Procedures

Now we assess the contribution of the CRP and SRP improvement procedures to the performance of MAPA. Table 4 presents results from additional tests with some instances involving 1-week and 4-week schedules. We conducted three independent experiments on the same set of initial solutions: 1) applying

CRP only, 2) applying SRP only and 3) applying both CRP and SRP. Table 4 presents the results of these experiments as follows. The initial solution cost is shown in column (InitCost), the cost obtained after applying CRP only to the initial solution is shown in column (CRP-Cost), the percentage cost reduction achieved by CRP is shown in column (%CRP), the cost obtained after applying SRP only to the initial solution is shown in column (SRP-Cost), the percentage cost reduction achieved by SRP is shown in column (%SRP), the cost obtained by applying both CRP and SRP to the initial solution is shown in column (CRP&SRP) and the percentage cost reduction achieved by applying both CRP and SRT is shown in column (%CRP&SRP).

Table 4: Contribution of CRP and SRP improvement procedures to the overall cost reduction in the improvement phase.

$\|N\|$	$\|D\|$	File	InitCost	Experiment 1		Experiment 2		Experiment 3	
				CRP-Cost	%CRP	SRP-Cost	%SRP	CRP&SRP	%CRP&SRP
25	7	1	343	309	9.91	313	8.74	307	**10.49**
50	7	1	1,123	580	**48.35**	584	47.99	580	**48.35**
75	7	1	939	880	**6.28**	882	6.07	880	**6.28**
100	7	1	2,476	1,289	**47.94**	1,292	47.81	1,289	**47.94**
30	28	1	3,998	1,583	60.40	2,149	46.24	1,573	**60.65**
60	28	1	6,267	3,186	49.16	3,364	46.32	3,184	**49.19**

Table 4 shows that CRP obtained more cost reductions over the initial cost than SRP. On some instances, CRP alone achieved the same improvement as when applying both procedures. However, Table 4 shows that overall, applying the two procedures achieves better results than applying either CRP or SRP alone.

5.5 Performance of MAPA

MAPA has shown to perform better on problem instances of larger size. Figure 6 shows a curve of %GAP for cost reduction and a curve of %GAP for soft constraint violations reduction for different problem instance sizes. Each point in the curves corresponds to the percentage of the average difference between the results obtained by MAPA and those reported in NSPLib. For example, the first point to the left in Figure 6 (a) indicates that on the problem instances with 7-days scheduling period and 25 nurses, MAPA obtained an average solution cost 0.37% higher. The last point to the right on Figure 6 (a) indicates that on the problem instances with 28-days scheduling period and 60 nurses, MAPA obtained an

average solution cost 2.93% lower. Figure 6 (b) shows similar information but with respect to the difference in soft constraints violations. For example, the two first points to the left indicate that on the problem instance with 7-days scheduling period and 25 or 50 nurses, MAPA obtained an average solution with the same penalty violations as those reported in NSPLib, The last point to the right of Figure 6 (b) indicates that on the problem instances with 28-days scheduling period and 60 nurses, MAPA obtained an average solution with 3.98% less soft constraints violations.

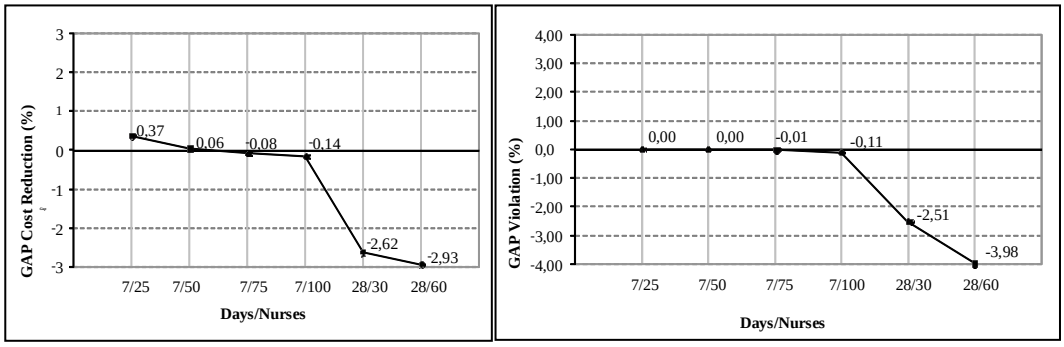

Figure 6: Comparing (a) average cost reduction and (b) average constraints violations difference between results obtained by MAPA and reported in NSPLib. A point below 0 indicates MAPA achieves better average results on that problem instance.

Figure 7 shows the percentage number of times that the best solution cost is reported in NSPLib, is obtained by MAPA or both. It can be seen that MAPA performs better as the size of instances grows.

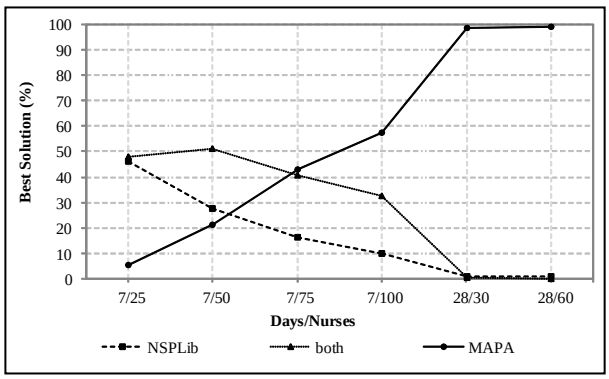

Figure 7: Percentage number of times that the best solutions are reported by MAPA and NSPLib.

Figure 7 shows that for instances with 7-days scheduling period and 50 nurses, the best results percentage achieved by MAPA and those reported in NSPLib are very close, 21.26% and 27.52%, respectively. However, MAPA

overcomes the results reported in the NSPLib for instances with 7-days scheduling period and 75 nurses.

These results show again that, as the size of instances grows with respect to the length of the scheduling period or the number of nurses, the performance of MAPA with respect to the solution quality improves considerably producing better results than those reported in NSPLib.

Although MAPA uses some more computational time compared to the results reported in NSPLib, the proposed algorithm can still be considered efficient for large instances. For example, producing a high-quality schedule for a problem with 4-week scheduling period and 60 nurses takes MAPA around 450 seconds (around 7.5 minutes) which can be considered practical.

5.6 Usability of MAPA

We should note that while it is common for heuristic algorithms (particularly meta-heuristics) to use randomization, MAPA is deterministic and hence multiple executions always generate the same results for the same input. In hospitals it is usually the case that nurse re-scheduling is required due to changes in demand, staff availability, etc. Another interesting aspect of MAPA is the possibility of using it for re-scheduling when facing unforeseen changes. Such re-scheduling is possible by applying the algorithm from the day in which the change happened onwards, while the previous days (left-hand side of the multipartite graph) are treated as historical records. Then, the multipartite model and multi-assignment procedure in MAPA is a suitable re-scheduling approach. The above features can be seen as very valuable for a heuristic approach to be accepted by human decision-makers (Cordeau et al. 2002) and particularly in the context of real-world healthcare environments (Petrovic and Vanden Berghe 2012).

6 Conclusions

In this work we proposed MAPA (multi-assignment problem algorithm) as a deterministic and effective heuristic algorithm for tackling a nurse scheduling problem. The proposed algorithm is based on an exact solution procedure with polynomial time complexity that solves a series of sub-problems (assignment problems). Each sub-problem corresponds to the assignment of shifts to all nurses

on a particular day, while considering the assignments already made on other days of the scheduling period.

We believe that MAPA satisfies the various desirable criteria defined by Cordeau et al. (2002) for heuristic methods. The *simplicity* criterion is met because the proposed algorithm does not require parameter tuning and it uses a classical well-known assignment problem which is easily solved. The *flexibility* criterion is also observed when incorporating new constraints which can be achieved by just introducing new values on the cost matrix (through equation 9) and modifying the appropriate *constraints update* procedure in the improvement stage of the algorithm. Reasonable *accuracy* and *speed* criteria are also observed in MAPA, particularly for larger problem instances, as it was shown in the experimental results of Section 5.

We also believe that MAPA satisfies several of the seven criteria proposed by Petrovic and Vanden Berghe (2012) for nurse scheduling methods. MAPA has good *expressive power* given its ability to tackle a wide variety of constraints by only modifying the procedure to construct the cost matrix. MAPA has good *flexibility* because the multi-assignment procedure can be easily adapted to different nurse scheduling scenarios. The results presented here also show that MAPA has good *algorithmic power* in terms of effectiveness and efficiency. MAPA has good *rescheduling* capability (as discussed in section 5.6) given the underlying multipartite model and associated multi-assignment procedure. MAPA is also good on *parameter tuning* because its performance does not depend on such process. MAPA meets the *maintenance* criterion because updating the domain knowledge about the specific nurse rostering problem being solved can be done easily by having a procedure to check each constraint (hard or soft) in order to construct the cost matrix. The only criterion of those proposed by Petrovic and Vanden Berghe that is not fully met by MAPA is the *learning capability* since the method is not capable of self-improving its performance over time.

In general, the solutions obtained by MAPA are better than the solutions reported in the NSPLib dataset. Taking into account all 248,602 solutions, MAPA obtained better solutions in 34.70% of the instances. On the opposite, NSPLib reports better solutions than those obtained by MAPA in 27.03% of the instances. Also, MAPA produced more feasible solutions than those reported in NSPLib. Therefore, we believe that this paper contributes with the introduction of a new

deterministic and effective heuristic algorithm to tackle the nurse scheduling problems in NSPLib. The paper also contributes by reporting new best results on some NSPLib instances compared to those by Maenhout and Vanhoucke (2007) obtained with different meta-heuristic approaches including the Electromagnetic method, Scatter Search and Genetic Algorithms.

As future research work, we suggest to investigate extensions to MAPA by considering new improvement procedures in addition to those described here. Also, it would be interesting to investigate the applicability of MAPA to other nurse scheduling benchmark datasets. Another suggestion is to combine the improvement procedures (CRP, SRP and perhaps others) with some meta-heuristic techniques to develop a hybrid approach. Having more improvement procedures, could allow using them as neighbourhood search routines and possibly to combine them into a VNS (variable neighbourhood search) style meta-heuristic. Also, the improvement of the computational time used by MAPA in larger problem instances is subject of future investigation.

Acknowledgements

The authors thank the referees very much for their valuable input during the review process which helped a lot in improving earlier versions of this paper. Also, thanks to the CNPq (National Council for Scientific and Technological Development) and CAPES (Brazil's Ministry of Education) for partial financial support to develop this work.

References

Bilgin, B., De Causmaecker, P., Rossie, B., Vanden Berghe, G.: Local search neighbourhoods to deal with a novel nurse rostering model. In: Proceedings of the 7th international conference on practice and theory of automated timetabling, pp. WA3-2/ 6-WA3-2/ 24. Montreal (2008)

Burke, E. K., Curtois, T., Post, G., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. Eur. J. Oper. Res. 188, 330–341 (2008)

Burke, E. K., De Causmaecker, P., Vanden Berghe, G., Van Landeghem, H.: The state of the art of nurse rostering. J. Sched. 7, 441-499 (2004)

Burke, E. K., Li, J., Qu, R.: A Pareto-based search methodology for multi-objective nurse scheduling. Ann. Oper. Res. 196(1), 91-109 (2012)

Carpaneto G., Toth P.**:** Primal-dual algorithms for the assignment problem. Discret. Appl. Math. 18, 137-153 (1987)

Cheang, B., Li, H., Lim, A., Rodrigues, **B**.: Nurse rostering problems - a bibliographic survey. Eur. J. Oper. Res. 151, 447-460 (2003)

Cordeau, J. F., Gendreau, M., Laporte, G. Potvin, J. Y., Semet, F.: A guide to vehicle routing heuristics. J. Oper. Res. Soc. 53**,** 512-522 (2002)

De Causmaecker, P., Vanden Berghe, G.: A categorisation of nurse rostering problems. J. Sched. 14(1), 3-16 (2011)

Haspeslagh, S., De Causmaecker, P., Schaerf, A., Stølevik, M.: The first international nurse rostering competition 2010. Ann. Oper. Res. Online: http://www.springerlink.com/content/f2u582476q346r66/ (2012)

Maenhout, B., Vanhoucke, M.: New Computational Results for the Nurse Scheduling Problem: A Scatter Search Algorithm. Lecture Notes in Computer Science, 3906, 159-170 (2006)

Maenhout, B., Vanhoucke, M.: An electromagnetic meta-heuristic for the nurse scheduling problem. J. Heur. 13, 359-385 (2007)

Maenhout, B., Vanhoucke, M.: Comparison and hybridization of crossover operators for the nurse scheduling problem. Ann. Oper. Res. 159, 333-353 (2008)

Osogami, T., Imai, H.: Classification of Various Neighborhood Operations for the Nurse Scheduling Problem. Lecture Notes in Computer Science 1969, 72-83 (2000)

Petrovic, S., Vanden Berghe, G.: A comparison of two approaches to nurse rostering problems. Ann. Oper. Res. 194(1), 365-384 (2012)

Vanhoucke, M., Maenhout, B.: NSPLib – A nurse scheduling problem library: a tool to evaluate (meta-)heuristic procedures. In: 31st Annual Meeting of the working group on Operations Research Applied to Health Services, pp. 151-165. Amsterdam (2005)